

一种基于二分编码的无线传感器与执行器网络移动覆盖算法

杜景林^{1,2,3} 郑若钦¹ 谢立^{2,3} 李娟¹

(南京信息工程大学电子与信息工程学院 南京 210044)¹

(南京大学计算机软件新技术国家重点实验室 南京 210023)²

(南京大学计算机科学与技术系 南京 210023)³

摘要 在无线传感器与执行器网络中,可以通过调整剩余节点的位置来提高目标区域覆盖率,以更好地为 sensor 节点服务。提出了一种基于二分编码的网络移动覆盖算法(SMR 算法)。在每次搜索中,试探性地调整失效节点区域的临近执行器节点位置来寻找最佳位置,重复此搜索过程直到覆盖数不再增加,以实现近似的局部最优覆盖。本算法提高了剩余节点的覆盖率,减少了执行器节点移动的消耗,与已有算法相比也表现出了较好的性能。

关键词 无线传感器与执行器网络,搜索式移动,网络覆盖度,节点失效

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.1.024

Mobile Coverage Algorithm Based on Dichotomy Coding for WSANs

DU Jing-lin^{1,2,3} ZHENG Ruo-qin¹ XIE Li^{2,3} LI Juan¹

(College of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China)¹

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China)²

(Department of Computer Science & Technology, Nanjing University, Nanjing 210023, China)³

Abstract In WSANs, the coverage of the target region can be enhanced by adjusting the location of remaining nodes to serve sensors better. A mobile coverage algorithm based on dichotomy coding was proposed. In each searching process, location of actor nodes near to the area of failure nodes can be adjusted in virtual to find the optimum position. This searching process is repeated until the coverage cannot be increased in order to reach the local optimum approximately. This algorithm improves the coverage rate of the remaining nodes and decreases the consumption of actors during moving. Compared to other existing algorithms, this algorithm has better performance.

Keywords WSANs, Searching move, Network coverage, Failure node

1 引言

在无线传感器与执行器网络(Wireless Sensor And Actor Networks, WSANs)中一个突出的挑战就是如何部署执行器(actor)节点来提高对传感器(sensor)的覆盖,该问题要求通过最少的能量消耗及包延迟来实现 actor 对 sensor 节点的最大覆盖。在 WSANs 中,为了执行单个或协作任务,actor 节点会移动到一定的区域内,而由于特殊场景或其他因素的影响,会出现一个或多个 actor 节点失效的情况。此时需考虑如何在保证剩余 actor 节点有效连接的前提下利用 actor 节点的移动来满足对最多数的 sensor 节点的服务,以等待失效节点的修复。而节点重定位策略^[1,2]和覆盖优化^[3]就成为了解决问题的关键。目前,关于 WSANs 在网络连接、覆盖与重定位机制等方面的研究已取得了一些研究成果。

Melodia 等人^[4]主要解决在一个特定事件区域内选取合

适的 actor 节点对事件做出快速反应,提出了一种竞拍协议,该协议在最低能量消耗与最小延迟的条件下分布式地分配 actor 给重叠区域,然而连接问题没有被考虑。类似地, K. Akkaya 等人^[5]提出了既保证降低 sensor 到 actor 之间延迟又提高节点覆盖的方法。在该项工作中,其目标是最小化 sensor 数据传输时延以使得 actor 节点及时到达事发现场。初始时 actor 节点为了满足最大覆盖均匀地放置在检测区域内(最小化 actor 覆盖范围重叠区域),之后通过重定位 actor 节点,实现算法目标,同时保证 actor 节点的连接。

王汝传等^[6]提出基于多目标遗传优化的无线多媒体传感器网络节能覆盖方法,即建立网络最大和最小密度下的覆盖模型,确立合适的撒播节点数;以网络最大运行轮次和部署节点数为依据,建立初始个体。班冬松等^[7]提出了移动无线传感器网络 k 栅栏覆盖构建算法。该算法定义了 k 栅栏覆盖最小移动距离和问题,提出了 GBMS 问题的近似算法—CBGB

到稿日期:2014-02-26 返修日期:2014-07-09 本文受国家自然科学基金面上项目(60873026,61021062,61372128),江苏省科技支撑计划(工业)项目(BE2011195, BE2010178),江苏省高校科研成果产业化推进工程项目(JH10-3),国家“九七三”重点基础研究发展规划(2006CB303000,2009CB320705)资助。

杜景林(1974—),男,博士,副教授,主要研究方向为无线传感器网络,E-mail: jldu@nuist.edu.cn;郑若钦(1989—),男,硕士生,CCF 会员,主要研究方向为无线传感器网络;谢立(1942—),男,教授,主要研究方向为分布式系统;李娟(1991—),女,主要研究方向为无线传感器网络。

算法,可能量高效地构建栅栏。Zhang 等^[8]提出分布式节点密度控制算法,根据节点邻居和自身位置计算相互覆盖关系。车楠等^[9]提出了基于实时性约束的 actor 节点覆盖优化部署策略,该策略是基于 Voronoi 图的最大实时覆盖部署策略。但已有的工作主要单独考虑节点覆盖最大化或者移动消耗最小化,这并不能保证 WSNs 失效后能在最短的时间内恢复到最好的工作状态。

节点失效后的恢复是 WSNs 的研究热点之一,现有工作主要单独考虑针对单一覆盖目标的部署或重定位问题,缺少对失效恢复覆盖问题的研究。本文构建了一种基于二分编码^[10-12]的移动覆盖方法(SMR),主要通过调整失效节点临近区域内的 actor 节点位置使其对于 sensor 的整体覆盖最大。该算法主要分为 3 步:首先,分割 WSNs 区域,根据区域个数为区域编码;其次,在 actor、sensor 节点之间传递数据包;最后,比较覆盖范围寻求最佳位置,最后移动节点。

2 问题描述

2.1 覆盖问题

Gungor 等人认为^[13],WSNs 的实时性是指存在某个上界 Δ_{ea} ,从 WSNs 中任何一个事件的发生到所需动作得到实施之间的延迟均应小于 Δ_{ea} ,等价于存在某个上界 R_n ,只要 sensor 节点 s 到 actor 节点 a 的距离小于 R_n , a 就可以对 s 发现的事件实时响应,此时称 a 实时覆盖 s ,而上界 R_n 就是 actor 的行动半径。

假设二维平面上 WSNs 由 m 个 sensor 节点和 n 个 actor 节点组成,sensor 集合为 G_s ,actor 集合为 G_a 。以 p_i 表示 actor 节点 i 的位置坐标($1 \leq i \leq n$),分别以 R_a 和 R_c 表示 actor 节点的行动半径和通信半径, $R_c = 2R_a$,定义目标区域的实时覆盖数为 σ ,则实时覆盖最大化的问题可定义为:

$$\begin{aligned} \sigma_{\max} &= \max |\bigcup_{i=1}^n C_i| \\ C_i &= \{ \forall s_j | d(s_j, a_i) \leq R_a, s_j \in G_s \} \\ \text{约束条件: } &G(a_i, E) \text{ 是连通图。} \end{aligned} \quad (1)$$

式(1)中的 $d(s_j, a_i)$ 代表 sensor 节点 j 到 actor 节点 i 的欧氏距离, C_i 表示被 actor 节点 i 实时覆盖的 sensor 节点集合, $|\bigcup_{i=1}^n C_i|$ 代表所有 C_i 并集的大小,即目标区域内所有被实时覆盖的 sensor 节点的数量,max 即求最大值。在约束条件下, $G(a_i, E)$ 代表所有 actor 节点的网络拓扑图, $G(a_i, E)$ 是连通图意味着对于 G_a 中的任意节点都至少可以通过一条路径到达其余任意节点,即网络是连通的。

定义 1 对于某个给定 R_a 和 R_c 的 WSNs 网络,若 σ 在 $(p_1, p_2, p_3, \dots, p_n)$ 处取得最大值,则称 $(p_1, p_2, p_3, \dots, p_n)$ 为最优节点位置序列。WSNs 网络的覆盖问题即是对于任意给定的 R_a, R_c 的网络,求解它的最优节点位置序列。

定义 2 假设节点 p, q, r 是失效节点的邻居节点,对于某个给定 R_a 和 R_c 的 WSNs 网络,设函数 σ_{local} 为 actor 节点 p, q, r 的局部覆盖数,函数 σ_{local} 在 p, q, r 处取得最大值,则称 $(p_1, p_2, \dots, p_p, \dots, p_q, \dots, p_r, \dots, p_n)$ 为局部最优节点位置序列。

2.2 初始场部署及失效恢复

节点的初始部署如图 1 所示,actor 节点静止于每个矩形区域中心,sensor 节点以 actor 节点为中心均匀分布。

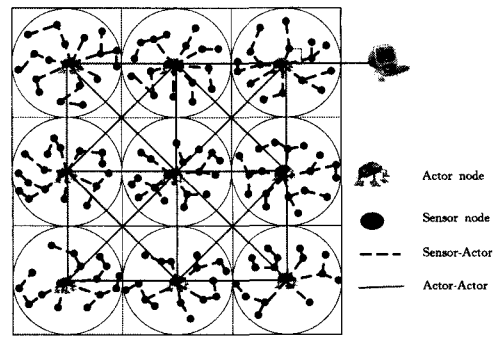


图 1 标准执行器分布图

在理想状态下,所有 actor 节点一直处于区域中心位置且正常工作,而所有 sensor 节点则一直处于与 actor 节点通信的状态。但在实际网络中,actor 节点并非总是处于静止且有效状态,当出现某些 actor 节点失效等极端情况,如图 2(上)所示,为保持网络连通性,部分或全部 actor 节点将在 R_c 范围内进行协同位移,感知孤立 sensor 群,去 sensor 孤岛,以实现任务区域内网络拓扑重构。

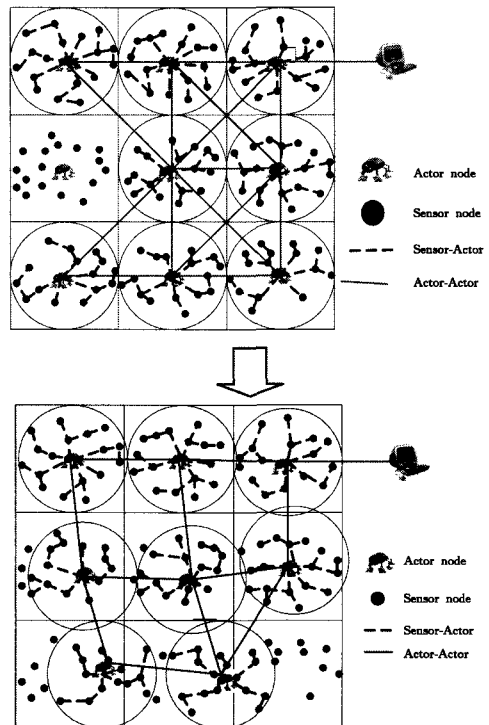


图 2 失效前后执行器分布图

图 2(下)描述了一个简单场景中节点失效之后的恢复情况,可以看出,失效节点的邻居会重定位到合适的位置,尽可能实现对更多 sensor 的覆盖。由于覆盖最大化问题是 NP-难问题,因此在节点失效后如何恢复覆盖是一个很大的挑战。

定理 1 WSNs 覆盖恢复问题是 NP-难问题

证明:覆盖恢复问题可以等价于求解局部覆盖最大化问题,因此先证明 WSNs 覆盖最大化问题是 NP-难的。假设 A 为求解 WSNs 覆盖最大化问题的算法,则利用 A 可以构造出求解能否用 k 个半径为 r 的圆盘完全覆盖平面点集 P 的判定问题的算法。在算法 A 中,判断 P 上的点是否被完全覆盖可以在多项式时间内完成。因此 k -圆盘完全覆盖问题可以在多项式时间内归约到 WSNs 覆盖最大化问题。由于 k -圆盘完全覆盖 P 已被证明为 NP-难问题,因此覆盖最大化问题

是 NP-难问题,从而证明了覆盖恢复问题是 NP-难问题。

如上所述,由于覆盖恢复问题是 NP-难问题,无法直接求解局部区域最大覆盖问题,因此本文利用 SMR 算法将网络分割成若干单元,从中选出近似的局部最优位置序列。

3 SMR 算法

SMR 算法是通过移动失效节点临近区域内的 actor 节点实现对 sensor 节点的整体覆盖最大化。SMR 算法分为 3 步: 1) 分割区域编码; 2) 数据包传递; 3) 搜索式移动, 即试探性移动节点比较覆盖范围, 寻求最佳位置。

3.1 二分编码基本原理

在移动恢复的过程中, actor 节点需要获取相应的 sensor 节点信息, 以更好地进行决策。为了网格化位置信息以便于计算, SMR 算法采用地址分配^[10-12]策略。SMR 的地址分配策略是将 actor 所在的分区等分为 4 个正方形分区, 再对各子分区进行四等分, 直到达到上限。各子分区都有对应的单元地址, 最小单元的分割情况如图 3 所示。经过此种编码, actor 可有效地通过单元地址获取 sensor 节点的相对位置信息, 为其移动提供参考。

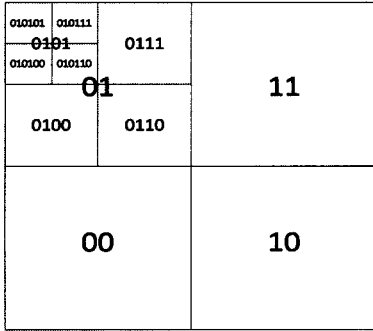


图 3 分割编码示意图

由于 sensor 节点不能使用编码空间的无限字母序列作为自己的单元地址, 因此可以通过 sensor 节点个数来选择合适的上限 k 作为反复等分的次数。 k 定义了算法的等分次数及单元地址长度, k 的范围是:

$$k \geq \log_2 \left(\frac{1}{\log_2 \frac{1}{n}} \right) \quad (2)$$

式中, n 代表 sensor 节点个数, k 取可能的最小整数值。任何单元的地址是包含编码空间元素有限序列, 单元地址定义如下:

$$\phi_{cell} = b_1 b_2 \cdots b_{2k-1} b_{2k}, b_i \in \{0, 1\} \quad (3)$$

SMR 算法单元计算策略描述如下:

算法 1

Input: $s(i), a(i), L$ // 分别表示 sensor 节点坐标, 单元中心坐标 (初始值是 actor 的初始坐标), 单元边长

Output: $\phi_{cell} = b_1 b_2 \cdots b_{2k-1} b_{2k}$

1. $j = 1$; // 用于记录地址下标

2. Dichotomy-coding($s(i), a(i), L, k$) // k 是迭代次数

3. While($j < 2k$)

4. $L = \frac{L}{2}$

5. // 根据坐标的相对位置判断地址取值

6. if $s(i), x > a(i), x$

7. $b_{j++} = 1, a(i), x = a(i), x + \frac{L}{2}$;

8. else

9. $b_{j++} = 0, a(i), x = a(i), x - \frac{L}{2}$;

10. end if

11. if $s(i), y > a(i), y$

12. $b_{j++} = 1, a(i), y = a(i), y + \frac{L}{2}$;

13. else

14. $b_{j++} = 0, a(i), y = a(i), y - \frac{L}{2}$;

15. end if

16. end while

算法 1 中每次等分都是对原有坐标进行折半查找 (纵横坐标各折半一次) 而产生一个 2bit 序列, 此序列作为当前单元地址的后缀, 扩展原有单元地址, 最终 sensor 所在单元地址是多次等分后产生的长度为 $2k$ 的地址序列。每个区域的 sensor 节点都在 actor 的通信半径之内, 所以 actor 节点可以轻易获得覆盖范围内的 sensor 节点信息。

为各单元编码后, 各 actor 节点可通过 sensor 节点编码了解其相对位置, 并通过本地计算了解各单元所包含的 sensor 节点个数, 同时各邻居 actor 交换各自区域的 sensor 分布情况。当出现失效节点后, 剩余节点需搜索并在比较中选取最优位置。为使各 actor 节点更好地了解本节点所处位置的 sensor 节点覆盖情况, 需在 actor 节点移动前对交换相邻的 actor 节点之间的数据进行协商。

3.2 节点移动算法描述

当出现失效节点后, SMR 算法会在二分编码的基础上求解近似局部最优位置序列, 然后由其邻居节点移动到这些位置上。

如上所述, 在二分编码之后, actor 可获得当前区域内的 sensor 节点信息, 失效节点的邻居将会以此为基础进行协商以确定最优候选位置。在 SMR 算法中, 邻居节点之间首先互相交换数据包以获得局部区域内的 sensor 节点分布情况, 之后所有邻居节点依次按轮搜索新位置并试探性移动 (即只是记录相关的候选位置和覆盖情况), 每轮只移动一个最小单元格, 每轮结束后广播信息给邻居节点计算覆盖情况。搜索依据是各编码单元内 sensor 个数及相对位置, 保证 actor 在每轮搜索后都向 sensor 节点密度高的方向移动。为了加快收敛速度, SMR 算法采用分粒度搜索策略, 先进行粗粒度搜索, 然后逐次细化, 即先考虑 ϕ_{cell} 的前两位编码代表的最大单元, 比如图 3 中如果 01 单元 sensor 节点最多就以此为基准进行更细粒度的搜索, 直到达到上限 k 。

为了减少覆盖重叠, 每次试探移动后会进行检测, 检测方法是利用数学上相切的概念及时修正各圆心的距离, 且利用排斥力的思想来修正原本相交的圆。各 actor 会在移动到与其余节点相切或者局部覆盖数不再增加的位置时 (说明向该方向移动不能进一步增加覆盖数了) 终止该搜索粒度下的移动, 等待更细的搜索。

在 SMR 算法搜索式移动策略描述中, $p()$ 代表节点坐标, Candidate-Position() 用于记录候选位置序列, 而 count 记录节点所覆盖的 sensor 节点数。算法描述如下:

算法 2

1. $\forall a_i \in \text{neighbour}(a_i)$ // 失效节点邻居

2. $j = 0$ // 用于记录试探移动的次数

3. Search in k round // 分不同粒度搜索

4. for a_i

5. dichotomy-search

6. end

7. for a_i not stop

8. j++
9. move to dense area //根据相对位置移动 a_i
10. detect circle and stop //检测到相切即终止移动
11. Candidate-Position(j)=update(Candidate-Position(j-1),p(a_i)) //更新第j次移动后的节点位置
12. Count(j) //记录第j次移动的局部整体覆盖数作为终止条件的判断
13. end
14. end searching
15. Find the Candidate-Position(m) assigned to maximum Count(j)
16. Move a_i to Candidate-Position(m)

在 SMR 算法返回一个局部最优节点位置序列后,各节点根据此位置序列进行真实移动。在 SMR 算法中,以 d 表示失效节点的邻居在移动过程中相互之间的距离,当 d 趋向于 $2R_a$ 时,两节点的覆盖重叠范围也就越小。

3.3 算法分析

定理 2 在 SMR 算法中,当移动比较次数趋于无穷大时,节点的位置序列会收敛于一个局部最优节点位置序列。

证明:令第 t 次移动时节点的位置序列为 $(p_1(t), p_2(t), \dots, p_p'(t), \dots, p_q'(t), \dots, p_r'(t), \dots, p_n(t))$, 节点的覆盖数为 $\sigma'_{p,q,r}(t)$, 节点 p, q, r 处于局部最优位置时的覆盖数为 $\sigma_{p,q,r}$ 。只需要证明当 $t \rightarrow \infty$ 时, $\forall p, q, r \in \{1, 2, \dots, n\}, \sigma'_{p,q,r}(t) \rightarrow \sigma_{p,q,r}$ 即可。

令 $\Delta\sigma(t) = (\sigma_{p,q,r} - \sigma'_{p,q,r}(t))_{1 \leq p,q,r \leq n}$ 。在每次搜索中,覆盖更多 sensor 节点的位置序列总会代替覆盖数较少的位置序列,可得到 $\forall t > 1, \Delta\sigma(t+1) \leq \Delta\sigma(t)$ 。因此,当 $t \rightarrow \infty$ 时, $\Delta\sigma(t) \rightarrow 0$, 可得 $\forall p, q, r \in \{1, 2, \dots, n\}, \sigma'_{p,q,r}(t) \rightarrow \sigma_{p,q,r}$, 证毕。

这证明了 SMR 算法通过从 8 个失效邻节点方向搜索最后能够渐进局部最优。而在 SMR 算法的迭代中,对于覆盖数的计算只需要知道 actor 节点的位置坐标、行动半径和移动方向即可。

4 仿真结果与分析

4.1 仿真环境与参数

SMR 算法的仿真是在 Visual Studio 2005 和 Matlab 环境下进行的。仿真参数设置如下:在一个 $1000\text{m} \times 600\text{m}$ 的区域内随机部署 300 个 sensor 节点,15 个 actor 节点,初始位置固定在矩形区域中心,各 actor 节点可以随时与其余节点通信。actor 节点的行动半径为 100m,通信半径为 200m,传输信道的传输率设定为 2Mbps。同一区域内的 sensor 节点之间可相互通信,相邻 actor 节点可直接通信。

仿真开始后,各 actor 节点会在所在区域内随机移动,移动后随机选取一个或多个失效节点。该实验中,采用选取平均值的方式进行算法仿真,以获得最公正的数据结果,置信度大于 90%,仿真结果也保持在样本均值的 3%~7%。

在节点失效后,使用 SMR 算法进行失效恢复,同时加入平均算法、COCOLA^[14] 进行性能比较。在 SMR 算法中,每个区域的编码计数器记录区域的 sensor 节点个数,每个 actor 节点中储存临近 actor 节点以及自身区域的覆盖 sensor 节点数。在 COCOLA 中,actor 形成了一个连通拓扑图。

算法的评价指标包含 actor 节点对 sensor 的覆盖数(考察恢复算法的效果),actor 节点的移动距离(考察能耗情况)以及 sensor 节点距离最近 actor 节点的最远距离(考察极端情况下 actor 对 sensor 的服务),将失效节点个数(分别取值

1,2,3)作为本次仿真的变量,以考察不同情况下的算法性能差异。

4.2 仿真实验结果

4.2.1 各算法分析

平均算法主要是在出现失效 actor 节点后,移动其余 actor 节点返回到初始位置。COCOLA 算法的重定位策略是算出每个区域的 1-中心位置,由 actor 节点移动到该位置。SMR 算法基于地址编码进行重定位,通过对 15 个矩形区域的分割编码计数,可清晰了解某区域的方位及涵盖的 sensor 节点个数,在 actor 节点移动策略中,可通过节点之间的数据交换简单完成移动后的覆盖统计。在产生失效节点时,失效节点的邻居进行临近移动,可最小化移动距离、时间,最大化局部覆盖数。

4.2.2 仿真结果分析

actor 节点对 sensor 节点覆盖数:由图 4 的仿真结果可知,在节点失效后,与其余两个算法相比,SMR 算法中 actor 覆盖了更多 sensor 节点。这说明,在实际网络环境中 SMR 算法可使全局网络在较为全面、受影响较小的情况下继续工作。

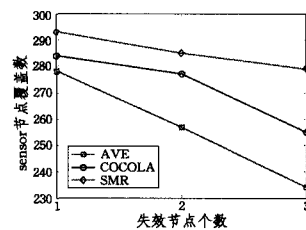


图 4 actor 节点移动后覆盖 sensor 节点数

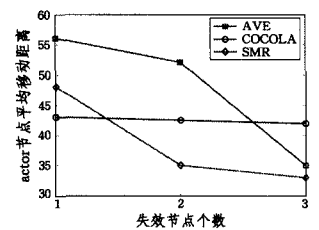


图 5 失效后 actor 节点平均移动距离

actor 节点的移动距离:在图 5 中,仿真结果表明 SMR 算法相比于其余两个算法在多数情况下所需移动距离都更小。这表明,SMR 算法最有可能在要求的时间内完成失效后的恢复工作,且移动所需能耗最小。图 6 则展示了单个 actor 节点的最大移动距离,可以看出 SMR 算法的单个节点移动极值小于其余两个算法(尤其是平均算法),这说明其在移动时的能耗负载平衡方面更有优势。

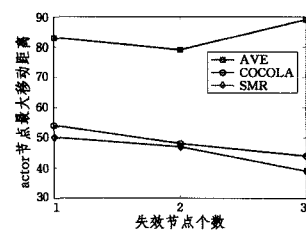


图 6 失效后 actor 节点移动的最大距离

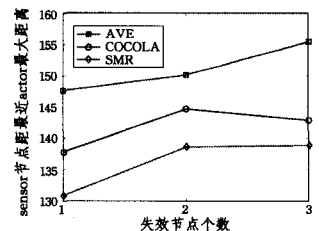


图 7 失效后 sensor 节点离最近 actor 节点的最远距离

sensor 节点与最近 actor 节点的最远距离:如图 7 所示,在任意多个节点失效情况下,SMR 算法总是比其余两个算法更优。这表明在 SMR 算法中未被覆盖的 sensor 节点也可以在更短的时间内被响应到。

从以上分析结果来看,在失效节点个数不同的情况下,SMR 算法相对于其余两个算法均可用相对较短的时间和较少的能耗部分恢复网络的工作,并使 actor 节点能够在最大范围内覆盖 sensor 节点,保证了正常的通信工作。

结束语 本文提出了一种基于二分法的搜索移动覆盖算

法 SMR,即通过对 WSANs 进行分割、编码、计数,使得 actor 可与邻居节点通信以获取其它区域 sensor 节点信息,通过算法求出失效区域附近的 actor 节点的近似最佳位置,然后进行移动。仿真结果表明,SMR 算法在保证 WSANs 正常工作的情况下,相对于现有算法增加了剩余 actor 节点对 sensor 节点的覆盖数,并减少了节点的移动消耗。

参 考 文 献

[1] Ren Y, Zhang S D, Zhang H K. Theories and algorithms of coverage control for wireless sensor networks[J]. Journal of Software, 2006, 17(3): 422-433

[2] Liu W, Cui L, Huang C C. EasiFCCT: A fractional coverage algorithm for wireless sensor networks[J]. Journal of Computer Research and Development, 2008, 45(8): 196-204

[3] Alaiwy M H, Alaiwy F H, Habib S. Optimization of Actors Placement within Wireless Sensor-Actor Networks[C]// 12th IEEE Symposium on Computers and Communications, 2007 (ISCC 2007). July 2007: 179-184

[4] Melodia T, Pompili D, Gungor V C, et al. A distributed coordination framework for wireless sensor and actor networks[C]// Proceedings of ACM Mobihoc. Urbana-Champaign, IL, May 2005

[5] Akkaya K, Younis M. COLA: A coverage and latency aware actor placement for wireless sensor and actor networks[C]// Proceedings of IEEE Vehicular Technology Conference (VTC). Montreal, CA, September 2006: 1-5

[6] 沙超,王汝传,黄海平,等.一种基于多目标遗传优化的无线多媒体传感器网络节能覆盖方法[J].电子学报,2012,40(1):19-26

[7] 班冬松,温俊,蒋杰,等.移动无线传感器网络 k-栅栏覆盖构建

算法[J].软件学报,2011,22(9):2089-2103

[8] Zhang H, Hou J C. Maintaining sensing coverage and connectivity in large sensor networks[J]. Ad Hoc & Sensor Wireless Networks, 2005, 1(1/2): 1-15

[9] 车楠,李治军,姜守旭. WSANs 中基于实时性约束的 actor 节点优化部署策略[J].软件学报,2011,22(6):1361-1372

[10] Houbadji T, Pierre S, Cast S. A distributed addressing and routing system for large scale wireless sensor and actor networks[J]. Computer Networks, 2009, 53(16): 2840-2854

[11] Doss R C, Chandra D, Pan L. Lease based addressing for event-driven wireless sensor networks[C]// 11th IEEE Symposium on Computers and Communications (ISCC'06). 2006: 251-256

[12] Zheng Y, Falko D. Dynamic address allocation for management and control in wireless sensor networks[C]// HICSS: 40th Annual Hawaii International Conference on System Sciences, 2007: 292

[13] Gungor V C, Akan O B, Akyildiz I F. A real-time and reliable transport (RT)₂ protocol for wireless sensor and actor networks [J]. IEEE/ACM Trans. on Networking, 2008, 16(2): 359-370

[14] Kemal A, Mohamed Y. Coverage and latency aware actor placement mechanisms in WSANs[J]. International Journal of Sensor Networks, 2008, 3(3): 152-164

[15] 温俊,蒋杰,方力,等.异构无线传感器网络的转发连通覆盖方法[J].软件学报,2010,21(9):2304-2319

[16] 孙超,尹荣荣,郝晓辰,等.异构无线传感器网络支配集拓扑控制算法[J].软件学报,2011,22(9):2137-2148

[17] 章韵,巨德文,陈志,等.基于可预测移动汇聚节点的无线传感网分簇算法研究[J].计算机科学,2012,39(6):89-92

[18] 李洪兵,熊庆宇,石为人.无线传感器网络非均匀等级分簇拓扑结构研究[J].计算机科学,2013,40(2):49-52,77

(上接第 95 页)

[15] 华夏渝,郑骏,胡文心.基于云计算环境的蚁群优化计算资源分配算法[J].华东师范大学学报:自然科学版,2010(1):127-134

[16] VMware Capacity Planner [EB/OL]. <http://www.vmware.com/products/capacity-planner/>, 2011

[17] IBM WebSphere CloudBurst [EB/OL]. <http://www-01.ibm.com/software/webservers/cloudburst/>, 2011

[18] 胡冷非.虚拟机 Xen 网络带宽分配的研究和改进[D].上海:上海交通大学,2009

[19] Tang Q, Gupta S K S, Varsamopoulos G. Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach [J]. IEEE Trans. Parallel Distribute System, 2008, 19(11): 1458-1472

[20] Moore J, Chase J, Ranganathan P, et al. Making scheduling "cool": temperature-aware workload placement in data centers, ATEC'05 [C]// Proceedings of the Annual Conference on USENIX Annual Technical Conference, 2005. USENIX Association, Berkeley, CA, USA, 2005: 5

[21] Buyya R, Ranjan R, Calheiros R N. InterCloud: utility-oriented federation of cloud computing environments for scaling of application services[C]// Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'10). Springer, Busan, South Korea, 2010: 13-31

[22] Wu L, Garg S. SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments[C]// 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2011

[23] Calheiros R, Vecchiola C, Karunamoorthy D, et al. The Aneka

platform and QoS-driven resource provisioning for elastic applications on hybrid Clouds[J]. Future Generation Computer Systems, 2012, 28(6): 861-870

[24] Vecchiola C, Calheiros R, Karunamoorthy D, et al. Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka[J]. Future Generation Computer Systems, 2012, 28: 58-65

[25] Calheiros R, Nadjaran toosi A, Vecchiola C, et al. A coordinator for scaling elastic applications across multiple clouds[J]. Future Generation Computer Systems, 2012, 28(8): 1350-1362

[26] Buyya R, Garg S, Calheiros R. SLA-Oriented Resource Provisioning for Cloud Computing: Challenges, Architecture, and Solutions[C]// Proceedings of the 2011 IEEE International Conference on Cloud and Service Computing (CSC 2011). 2011

[27] Nadjaran toosi A, Calheiros R, Thulasiram R, et al. Resource Provisioning Policies to Increase IaaS Provider's Profit in a Federated Cloud Environment[C]// 2011 IEEE International Conference on High Performance Computing and Communications (HPCC 2011). 2011: 279-287

[28] Calheiros R, Ranjan R, De rose C, et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms [J]. Software: Practice and Experience, 2011, 41(1): 23-50

[29] Buyya R, Ranjan R, Calheiros R. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit[C]// Challenges and opportunities, International Conference on High Performance Computing & Simulation, (HPCS'09). 2009: 1-11